First Hit
Search Forms
Search Results
Help
User Searches
Preferences
Logout
Generate Collection

DOCUMENT-IDENTIFIER: US 20020174207 A1
TITLE: Self-healing hierarchical network management system, and methods and apparatus therefor

Application Filing Date:
20010228

Detail Description Paragraph:
[0115] In summary, the Node Manager provides NMS interface and local node management, as well as providing signaling, routing and fault protection functions (all using the Node Manager's application software), provides real-time LCM provisioning, receives monitored parameters and alarms/faults from each LCM, aggregates monitored parameters and alarms/faults from each line card into a node-wide view, processes node-to-node communication messages, provides remote software download capability, distributes new software to all LCMs, is expandable to utilize a more powerful CPU (through plug-in processor 512), such as of RISC design, is built on a Real-Time Operating System (RTOS), provides intra-OTS networking support (e.g., LAN connectivity to LCMs), and provides node-to-node networking support.

L17: Entry 19 of 34 File: USPT Jun 12, 2001

DOCUMENT-IDENTIFIER: US 6247016 B1
TITLE: Decision tree classifier with integrated building and pruning phases

Application Filing Date (1):
19981110

Brief Summary Text (15):
FIG. 2 is an example of a decision tree for the training data in FIG. 1. Each internal node of the decision tree (denoted by a circle in FIG. 2) has a "test" involving an attribute, and an outgoing branch for each possible outcome. For example, at the root node 10 the test is "is the salary level of the applicant less than $20,000.00?" If the answer to this inquiry is "no," the loan application is automatically accepted, ending the inquiry and establishing a "leaf" 20 (a leaf is the ultimate conclusion of a partition after no further inquiry is to be made, and is denoted by a square in FIG. 2) for the acceptance. Thus, in the example, an applicant who has a salary greater than $20,000 is classified in a class for those applicants who qualify for a loan based on their salary alone.

Brief Summary Text (22):
Each node of the decision tree maintains a separate list for every attribute. Each attribute list contains a single entry for every record in the partition for the node. The attribute list entry for a record contains three fields--the value for the attribute in the record, the class label for the record, and the record identifier. Attribute lists for the root node are constructed at the start using the input data, while for other nodes they are derived from their parent's attribute lists when the parent nodes are split. Attribute lists for numeric attributes at the root node are sorted initially and this sort order is preserved for other nodes by the splitting procedure. Also a histogram is maintained at each node that captures the class distribution of the records at the node. Thus, the initialization of the root node in Step 1 of the build algorithm of FIG. 3 involves (1) constructing the attribute lists, (2) sorting the attribute lists for numeric attributes, and (3) constructing the histogram for the class distribution.

WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Thursday, February 12, 2004

Hide?	Set Name	Query	Hit Count
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L18	L17 and l10	0
☐	L17	L15 and l6	34
☐	L16	L15 and l1	0
☐	L15	20010515	341
☐	L14	(node or server or router) near8 tree near8 attribute	552
☐	L13	20010515	13
☐	L12	(adapting or adaptaiton) near8 (new or added) near8 (node or server or cache)	17
☐	L11	20010515	26
☐	L10	(new or adding) near8 (node or server) near8 (query or message) near8 (download or downloading)	64
☐	L9	(new or adding) near8 (node or server) near8 (query or message)	3266
☐	L8	20010515	6
☐	L7	L6 and l1	14
☐	L6	(initialize or initialization or initializing or adapt or adapting or adaptation) near8 (node or server or router or switch)	19105
☐	L5	20010515	3
☐	L4	(smart or intelligent) adj2 (cache or caching) adj2 server	4
☐	L3	L2 and (download or downloading)	12
☐	L2	20010515	23
☐	L1	(smart or intelligent) near8 (cache or caching) near8 server	43

END OF SEARCH HISTORY

First Hit

Generate Collection

L3: Entry 3 of 12 File: PGPB Sep 26, 2002

DOCUMENT-IDENTIFIER: US 20020138640 A1
TITLE: Apparatus and method for improving the delivery of software applications and associated data in web-based systems

Abstract Paragraph:
An improved system for streaming a software application to a plurality of clients comprises a principal server having the software stored thereon as a plurality of blocks and a plurality of intermediate servers between the principal server and the clients. The principal server is configured to stream program and data blocks to downstream devices in accordance with a dynamic prediction of the needs of those devices. The intermediate servers are configured to cache blocks received from connected upstream devices and service requests for blocks issued from downstream devices. In addition, the intermediate servers are further configured to autonomously predict the needs of downstream devices, stream the predicted blocks to the downstream devices, and if the predicted blocks are not present in the intermediate server cache, request those blocks from upstream devices. The intermediate servers can also be configured to make intelligent cache purging decisions with reference to the contents of the caches in other connected devices.

Application Filing Date:
20001222

Summary of Invention Paragraph:
[0004] The Internet, and particularly the world-wide-web, is a rapidly growing network of interconnected computers from which users can access a wide variety of information. Initial widespread use of the Internet was limited to the delivery of static information. A newly developing area of functionality is the delivery and execution of complex software applications via the Internet. There are two basic techniques for software delivery, remote execution and local delivery, e.g., by downloading.

Summary of Invention Paragraph:
[0007] In a local delivery embodiment, the desired application is packaged and downloaded to the user's computer. Preferably, the applications are delivered and installed as appropriate using automated processes. After installation, the application is executed. Various techniques have been employed to improve the delivery of software, particularly in the automated selection of the proper software components to install and initiation of automatic software downloads. In one technique, an application program is broken into parts at natural division points, such as individual data and library files, class definitions, etc., and each component is specially tagged by the program developer to identify the various program components, specify which components are dependent upon each other, and define the various component sets which are needed for different versions of the application.

Summary of Invention Paragraph:
[0008] Once such tagging format is defined in the Open Software Description ("OSD") specification, jointly submitted to the World Wide Web Consortium by Marimba Incorporated and Microsoft Corporation on Aug. 13, 1999. Defined OSD information can be used by various "push" applications or other software distribution

environments, such as Marimba's Castanet product, to automatically trigger downloads of software and ensure that only the needed software components are downloaded to the client in accordance with data describing which software elements a particular version of an application depends on.

Summary of Invention Paragraph:
[0009] Recently, attempts have been made to use streaming technology to deliver software to permit an application to begin executing before it has been completely downloaded. Streaming technology was initially developed to deliver audio and video information in a manner which allowed the information to be output without waiting for the complete data file to download. For example, a full-motion video can be sent from a server to a client as a linear stream of frames instead of a complete video file. As each frame arrives at the client, it can be displayed to create a real-time full-motion video display. However, unlike the linear sequences of data presented in audio and video, the components of a software application may be executed in sequences which vary according to user input and other factors.

Summary of Invention Paragraph:
[0012] One challenge in implementing a predictive streaming system is maintaining an acceptable rate of data delivery to a client, even when many clients are executing streaming applications. A technique which has been used to improve the delivery time of Internet hosted data accessed by many users is to use caching techniques. In standard Internet-based web-page distribution systems, caching systems are linked between a primary server hosting the web site and the end users or clients, with each cache server servicing a number of corresponding clients. These cache servers are used to store web pages that have been requested by a client from a principal server. Each time a client requests a particular web page, the request is processed by the respective cache server which is servicing the client. If the requested page is present in the cache server, the page is extracted from the cache and returned to the client. If the requested page has not been previously accessed by any of the clients corresponding to the particular cache server, the cache server forwards the request to the primary server to download the page from the Web site, stores the retrieved web page, and serves that page to the client.

Summary of Invention Paragraph:
[0015] The present invention relates generally to a method and system for improving the delivery of software applications and associated data, which can be stored in databases, via a network, such as the Internet. One or more intermediate tiers of intelligent caching servers are placed between a principal application server and the streaming application clients. The intermediate servers store streamed blocks, such as software application modules or streamlets and other database modules, as they are transmitted from the principal server to a client. As a result, further requests by the same client or other clients associated with the intermediate servers for previously stored information can be streamed from the intermediate servers without accessing the principal server.

Detail Description Paragraph:
[0060] As discussed above, various techniques can be used to predict the order in which a client will require various program elements during execution. The following is a more detailed discussion of a particular technique for predicting this order for use in determining an order in which program elements should be streamed to a particular client. As discussed above, this information can be used in the presently disclosed system, along with additional information, such as the contents of related intermediate servers, download times, etc., by each intermediate server to determine the most appropriate streaming sequences to downstream devices and to further determine program elements which should be requested from upstream devices in anticipation of future needs.

Detail Description Paragraph:

[0062] To minimize module download delays experienced by a user, module "E" may be transparently streamed from a server to the client computer before it is required at the client. Transparent streaming allows future module use to be predicted and modules to be downloaded while other interrelated modules "A" are executing. Referring to FIG. 6, the execution order of application modules "A" through "H" can be visualized as a directed graph 600 rather than a linear sequence of modules. For example, as illustrated by the graph, after module "A" is executed, execution can continue at module "B," "D," or "E." After module "B" is executed, execution can continue at module "C" or "G." The execution path may subsequently flow to additional modules and may return to earlier executed modules.

WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Thursday, February 12, 2004

Hide?	Set Name	Query	Hit Count
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L5	20010515	3
☐	L4	(smart or intelligent) adj2 (cache or caching) adj2 server	4
☐	L3	L2 and (download or downloading)	12
☐	L2	20010515	23
☐	L1	(smart or intelligent) near8 (cache or caching) near8 server	43

END OF SEARCH HISTORY